# Integrating Actuator Fault and Wheel Slippage Detections within FDI Framework

NAIM SIDEK, NILANJAN SARKAR
Department of Mechanical Engineering
Vanderbilt University
Robotics and Autonomous Systems Laboratory
Station B, 351592, 2301 Vanderbilt Place,
Nashville, TN. 37235-1592, USA
naim.sidek@vanderbilt.edu, nilanjan.sarkar@vanderbilt.edu
http://robotics.vuse.vanderbilt.edu

*Abstract:* - We have witnessed a significant advancement in the field of mobile robot applications in the past two decades. From performing mission critical tasks such as in planetary exploration to simply doing household chores, this type of robots requires availability, reliability and safety of its operations. Consequently, there is a growing demand for fault tolerant control system (FTCS) for mobile robots where one of it major component is the fault detection and isolation (FDI) module. In our FDI study, one of the challenges in designing the robot model is to create an accurate and robust dynamics model. In this paper, the environment structure, which is the dynamic change in the surface characteristics, will be included. We will utilize Local Neural Model (LNM) in our robot modeling process. Wheel slippage is an event happened when the robot wheels lose their full grasp of the surface. This may cause the robot to deviate from its desired trajectory. On the other hand, the fault on the robot actuator may also lead to similar behavior but need different solution. In this paper we will expand the functionality of FDI module to infer a real actuator fault and wheel slippage in which, the solutions for both events can be combined in a single, common controller.

*Key-Words:* - mobile robot, actuator fault, wheel slippage, local neural model, fault tolerance, fault detection and isolation

## 1 Introduction

We have witnessed a significant advancement in the field of mobile robot applications in the past two decades. There are numerous practical situations where WMR is proven to be useful [4,6,10] that range from performing mission critical tasks such as in planetary exploration to simply doing household chores. However, the complexity of the robots has also increased in order to accommodate the demands of sophisticated tasks. Most modern robots include multiple actuators, numerous sensors and other components. As a result of this increased complexity, the possibility of developing system faults is also increased. If this problem is not solved properly, it may lead to catastrophic consequence especially for autonomous systems. Hence, there is a growing need for the system to have a mechanism to handle the faults. This requires the system to detect and isolate faults, and reconfigure itself automatically as soon as the fault occurs. Fault tolerant control system addresses the above-mentioned issues to ensure the availability, reliability and safety of the system operations.

One of the major components in FTCS is fault detection and isolation module. In the literature, the approaches to solve the FDI problem are broadly classified into two classes: model-based and model-free approaches. Model-based fault detection approach utilizes the mathematical model of the plant to generate residuals. Residuals are the measure of discrepancy between the expected and the measured system behaviors. The problem of model-based FDI method is that it requires an accurate mathematical model to reduce the generation of inaccurate residuals. This is required to avoid false alarms in detecting the faults. This problem becomes more critical when the robot is subjected to unknown environmental changes or external disturbance (e.g., changes in road conditions and/or surface characteristics). Model-free methods are likely to be more successful in avoiding this issue because instead of depending on the model, they seek to learn the system behaviors. Neural network (NN) is a model free technique that has been widely used to replicate the dynamics of robotic systems [5,8,10,19]. Its learning capability can capture modeling uncertainty inherent to the robot dynamics. In this research, we use Local Neural Model to learn not only the system dynamics that is inherent to the nominal robot operation but also the effect of environmental changes on its behavior. LNM has the ability to be trained and can adapt to the change of its parameters faster due to its linear optimizing scheme.

In this work, we consider wheel slippage as an effect to the change in surface characteristics, which consequently affects the nominal robot behavior. It is an event that happened when the robot wheel loses its full grasp of the surface. Traction controller is commonly employed to correct this problem to avoid the robot from deviating from its desired trajectories. However, if an actuator fault occurs, that too will affect the nominal behavior of the robot. While the changes in nominal behaviors due to these two different events might be similar, each situation calls for a different solution. In this paper we develop an FDI module that can differentiate between an actuator fault and wheel slippage. We first present the framework in Section 2. We then describe the experimental set-up in Section 3. The results are finally discussed in Section 4.

## 2 Neural Modeling of the Mobile Robot

The model of our mobile robot composed of two wheel systems. Hence, we try to describe the relationship between the speed of the wheels and their torque-command voltages. Most commercial mobile robots allow a very limited access to their hardware and electronics. As a result, to measure the torque-command voltage in the motor is difficult without need to do some major hardware modification. On the other hand, we note that many robots have servo-controllers that are fixed and embedded in the systems' control feedback loops. Thus the robot dynamics can generally be described as:

$$\begin{aligned} &y(t) = f\left(u(t), x(t), x'(t), \ldots, x^{n-1}(t), y'(t), y''(t), \ldots, y^{n-1}(t)\right) \\ &x(t) = f\left(e(t)\right) \\ &x'(t) = f\left(e(t), e'(t)\right) \end{aligned} \tag{1}$$

where $y, e$ and $e'$ are the wheel linear speed, error in linear speed and the derivative of the error. These are the parameters we used to describe the system relationship in the neural network model. In addition, the $u$ and $x$ are the input and the state of the system, respectively. $n$ is the order of the system. The equation shows a nonlinear relationship that arises from the nonlinear motor torque characteristic as well as the motor static friction. However, the overall behavior of the mobile robot is not just the summation of individual wheel behaviors. A direct cross coupling relation between the robot's two wheels also needs to be realized. For instance, if the commanded torque to one of the wheels is zero, the static friction of that wheel will directly affect the other wheel. The same effect applies, when the robot wheels are not moving at the same velocities. The faster or slower wheel will force the other wheel to move faster or slower.

There have been a number of model-free methods reported in the literature that have been applied to learn the behaviors of various robotic systems. In [2], the author reports the use of fuzzy-neural network to approximate the potential based guiding model for mobile robots. In [3,6,7], the robot model is generated through system identification using Takagi-Sugeno (TS) fuzzy system. The idea of using local linear model tree (LOLIMOT) to model nonlinear system has also been reported. [4,5] report the use of LNM, which resembles the first order TS fuzzy model or general radial basis function (RBF) network for their nonlinear robot system. Meanwhile [8,9,10] report the employment of a single multilayer perceptron (MLP) network to model a variety of nonlinear robot systems.

In this paper, a set of LMN is utilized in the system modeling. This method suggests the division of the operating region, where each LNM will approximate the dynamics of the respective region. This is described as:

$$y = F(v, w) = \sum_{i=1}^{M} f_i(v) p_i(w) \tag{2}$$

where $M$ is the number of local models, $f(v)$ is a linear functions of the measurement vector $v$, and $p(w)$ is a Gaussian function in RBF network. In our case the input vector, $w$ to the network consists of the difference between the desired velocity and the actual velocity, $e$ and its previous value. The output, $y$ is the wheel speed. Thus, each local model will estimate the speed of the wheel given the error and previous error values. Our $i$th local model then can be represented and simplified to the following discrete equation form,

$$y_i(k) = f_i\left(e(k), e(k-1)\right) \tag{3}$$

A switching mechanism is built to determine the appropriate local model to be employed during the operation based on the measurement of current wheel speed. The generation of local models is based on the division of the robot-operating region. In order to do that, we utilize a grid partitioning techniques in which the range of the wheel speed is divided into several equally spaced intervals. We also have two-set of main networks representing two-set of robot wheel systems.

In our FDI study, the occurrence of actuator fault can be generalized based on the robot trajectories. In doing so, once the fault is identified, depending on the level of fault, the next robot path can be decided. Thus, our LNM are classified into three types of behaviors, namely moving straight, small turning and big turning as depicted in the

Fig. 1 and 2. These behaviors are chosen because they can be used to traverse most complex trajectories.

In the next design stage, each of the two inputs and one output LNM is trained based on random training set or a priori knowledge captured during nominal robot operations.

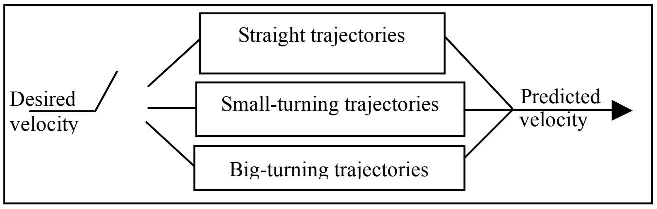


Fig. 1: LNM classification based on the behavior of robot trajectories

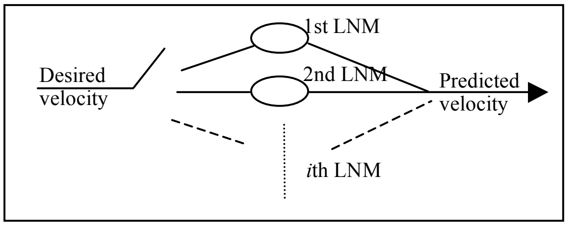


Fig. 2: LNM for straight trajectory

### 2.1 Health and safety of mobile robot

Actuator fault can comes in a variety of forms. For mobile robot, the actuator fault (motor fault) is happened to be the most common fault. Motor faults are generally due to several causes such as wrong brush contact, bearing faults, performance deficiency, electrical faults and stator iron core faults [17]. As we are looking on the actuator faults or motor faults on the wheels, monitoring for the right parameter is crucial. Wheel speed seems to be an appropriate parameter to be monitored to detect actuator fault.

Wheel slippage on the other hand, can affect the position estimation error and consequently loss of traction. By minimizing slippage, the odometric errors will be minimized and this will reduce the overall energy consumption and increase robot traction performance [11]. A number of current methods that aim at detecting slippage have been reported. [12] uses a system based on fusing visual odometry and inertial measurements through a Kalman filter pose estimator implemented on mars rover. [15, 18] report that by classifying the visual traction, the slip factor can be associated to the predefined values of various terrains. In [13], the author reports the use of sun sensor to improve the robot state estimates while in [14], the authors describe the use of combination visual odometry and sun sensor. By comparing the encoder measurement, gyro and electric motor current, [16] reports a technique to measure slippage for mobile robot successfully. For a robot to have the ability to provide substantial safety countermeasures, a number of possible health and safety indicators need to be implemented as reported in [15]. Among others are component failure, available power, chassis altitude, wheel slippage and sinkage, and mechanics compliance. Many of these issues have been thoroughly researched and tackled, but apparently we find that each solution for a particular issue seems to be isolated from each other. This generally can contribute to the waste of resource on the robot system. In this paper, we will exploit the resemblance in robot behaviors when the two events namely wheel slippage and actuator fault occurs. For that matter, we want to expand the functionality of our FDI module so as to combine the solution for these healthy and safety issues. The goal is to simplify the solution by governing and optimizing the resource on the robot without sacrificing the system performance.

## 3 Experiments

### 3.1 General description

As described before, in detecting motor fault, the neural model of the two-set of wheels will be used as a reference healthy model. The actual velocity that may differ from the nominal velocity due to actuator fault is obtained from motor encoders. This velocity will be used to compare with the healthy (nominal) velocity obtained from the neural model. The computation is done in real-time. The errors between these two velocities are then fed into the decision-making block to identify the size, existence, source and cause of the fault. We are not going to touch on this issue in this paper.

The analysis of slip, on the other hand, is done based on the following equation,

$$\lambda = \left( \frac{V_w - V_r}{V_w} \right) \times 100 \tag{4}$$

where $V_w$ and $V_r$ are the velocities of the wheel and robot, respectively. $\lambda$ is the wheel slip ratio which is defined dimensionlessly as a percentage of robot forward velocity. When this value is nonzero, wheel slippage occurs. In the study of traction control, this value is regulated to optimize the robot traction performance. Apparently, the slip factor can be easily obtained if the two velocity parameters are observable.

For this research, the wheel velocity is available from the shaft encoder. However, it is always a challenge to measure the actual robot velocity due to the dynamic interaction between the wheel and the surface.

### 3.2 Hardware setup and experiment tasks

Since we are focusing on the robot running in the indoor environment, we use accelerometer to measure velocity. In order to obtain reliable robot velocity especially during wheel slippage, the accelerometer is mounted on top of the robot as close as possible to the wheel. This is shown in the Fig. 3. We use a 3-axis accelerometer, which features reconfigurable bandwidth up to 500 Hz and a measurable range up to +/- 2g.

The experiment is divided into two parts. The first part is to capture the robot model using LNM technique. For that, we define the operating region for each of the robot wheels to be from 0 to 1400mm/s (maximum possible velocity). We then equally divide this operating region into twenty-eight sections. This results in 28x28 possible input sections where each possible combination has distinct robot trajectories. By using all possible input combinations, we run the robot on the normal tiled office floor and capture data from the wheel encoder to train the LNM. The goal is to learn the healthy model of the robot. The final set of LNM is then grouped into three classes, which are: moving straight, small turning (turn radius is less then 1m) and big turning (turn radius is equal or bigger than 1m). In the second part we want to see the response and performance of the fault and slippage detection mechanisms, hence, we run a set of experiments following the strategy below:

i. First, run the robot on normal floor and capture the predicted, encoder-measured and accelerometer-measured velocity values. This will demonstrate whether those measure velocity values conform one another.

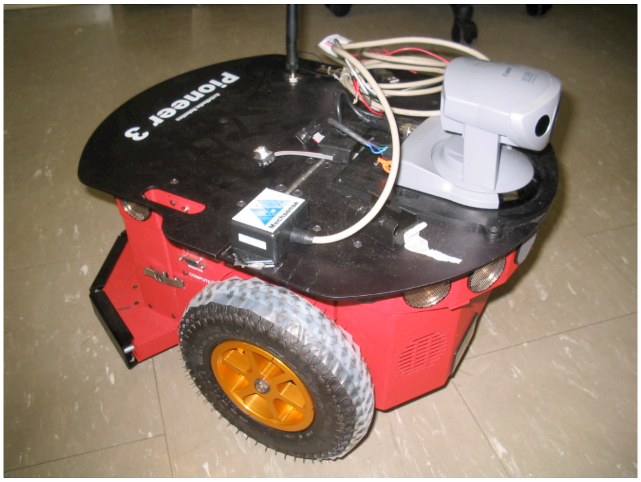


Fig. 3: Pioneer P3DX with mounted accelerometer

ii. Second, run the robot and introduce actuator fault at 3.5-second of robot trajectory. The idea here is to investigate whether the actuator fault can be detected by comparing the predicted and the encoder-measured velocities values. The fault is introduced in the form of a step signal, which represents power loss of the servo system. The following equation is the simple closed-loop servo system of the robot.

$$T(s) = \frac{y(s)}{x(s)} = \frac{K_c(s)K_p(s)}{1 + H(s)K_c(s)K_p(s)} \tag{5}$$

We model the fault by amplifying the desired input, $x(s)$ by 1.1%. The form of the system response approximates the form of response in which the actuator fault such that as much as 10% of power loss in servo system occurs. Since the sensor and actuator are serially connected, the effect of the actuator fault is reflected in the sensor term, $H(s)$.

iii. Third, run the robot on a slippery surface (lubricated vinyl floor) to induce wheel slippage and investigate whether it can be detected by the accelerometer.

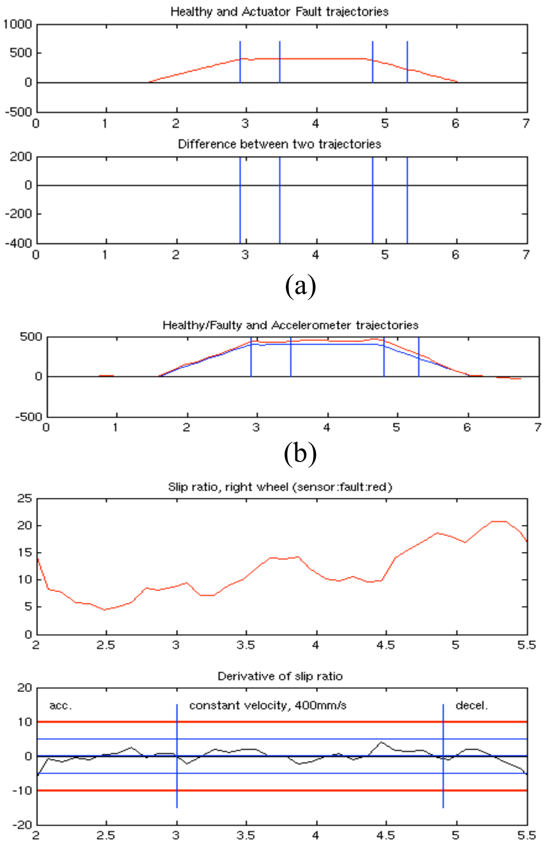
For every experiment, we apply the same desired robot velocity, which is set to be 400mm/s.

## 4 Results And Discussions

In the first experiment, we run the robot without introducing either an actuator fault or slip. Fig.4a shows that no significant deviation between the neural model predicted trajectory and the encoder measurement based trajectory. Thus, this result validates that the trained model sufficiently captures the robot behavior. In Fig. 4b, we present results that show that the trajectories generated from the encoder data and from the accelerometer data match reasonably well when there is no fault or appreciable slip. In this case, however, note that from Fig.4c, that the slip is within 20% and its derivative is bounded within +/-5, which we considered to be a safe range of operation. We defined the above bounds of the slip derivative as our threshold of detecting unacceptable slip.

For the second experiment, we introduced an actuator fault after the robot achieved its desired velocity. As shown in Fig.5a, the fault happened at 3.5-second. This actuator fault was to signify 10% loss of motor power of the right wheel. As expected, the difference between the healthy and the real robot trajectories grew from this point onwards. We can see that the accelerometer signal still tracks the robot trajectory excellently as shown in Fig.5b. Fig.5c shows that the derivative of slip ratio still falls within the safe bound during the period of constant robot velocity.

In the third experiment, the robot is commanded to move on a lubricated floor to reduce the friction between the wheel and the surface. As the encoder shows no sign of slip reflected in Fig.6a, the accelerometer is able to detect slip. The plot in Fig.6b shows the decrease of robot velocity as the slippage occurs. Fig.6c confirms the slip as the signal clearly goes out of bound.

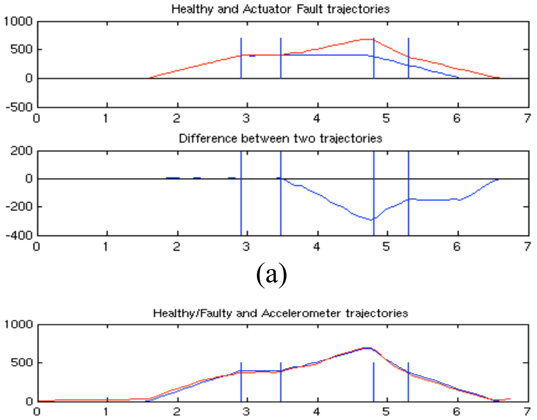


(a)

(b)

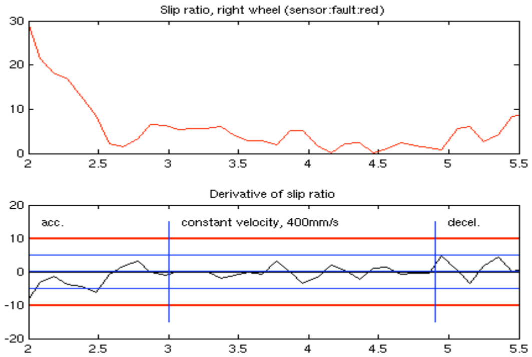
(c)

Fig. 4 (a),(b),(c): Experiment without motor fault and wheel slippage



(a)

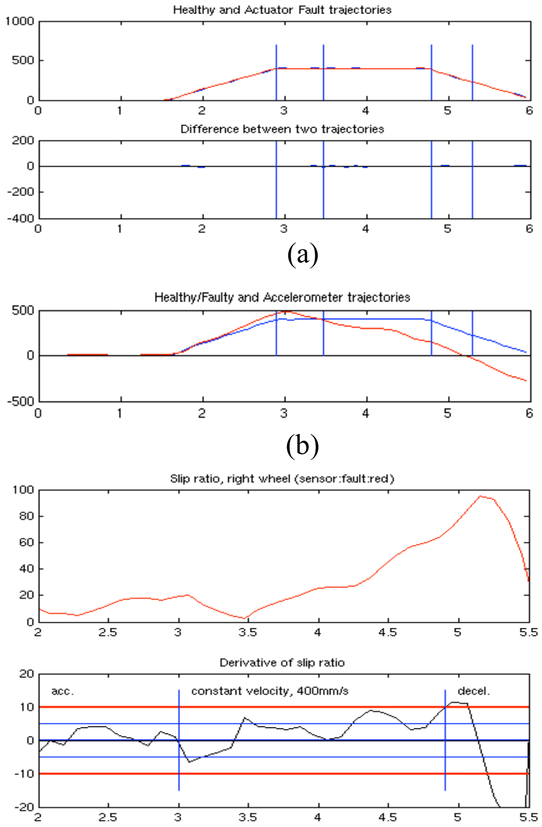
(b)



(c)

Fig. 5 (a),(b),(c): Experiment without motor fault but wheel slippage

From these experiments, we can see that the fault can be reliably deduced from the difference between the output of neural model and the encoder reading. We also see that by using accelerometer, the velocity of the robot can be tracked, hence the wheel slippage can be detected.



(a)

(b)

(c)

Fig. 6 (a),(b),(c): Experiment without motor fault but wheel slippage

## 5 Conclusions And Future Works

This paper introduces a new way to optimize resource on mobile robot where a substantial safety countermeasure is implemented. Among major concerns on robot healthy and safety are actuator fault and wheel slippage. The result from this paper demonstrates that the use of LNM technique to model the mobile robot is quite successful. It also show that the manipulation of neural predicted, encoder-measured and accelerometer-measured velocity values could be used to deduce actuator fault and wheel slippage. In the next stage of this research, this information will be used to help the design of common controller for the FDI system. The classification of the actuator fault based on the trajectory behaviors may also add another parameter to the process of designing the controller.

*References:*

[1] http://magic.uni-duisburg.de/mc/general.htm

[2] Yasuhiko Dote, Seppo J. Ovaska, Industrial Applications of Soft Computing: A Review, *Proc. Of The IEEE*, Vol. 89, No. 9, 2001, pp. 1243-1265

[3] Yixin Diao, Kevin M . Passino, Intelligent fault tolerance control using adaptive & learning methods, *Control Engine. Practice*,Vol.10,No.8,2002,pp801-817

[4] EN Skoundrianos, SG Tzafestas, Fault diagnosis on the wheels of mobile robot using LMN Network, *IEEE Robotics & Automation Magazine,* Vol.11, Issue 3, 2004, pp. 83-90

[5] EN Skoundrianos, SG Tzafestas, Fault diagnosis via local NN, *J. of Mathematics & Computers in Simulation* Vol.60, No.3-5, 2002, pp.169-180

[6] Ichtev A., Hellendoorn J., Babuska R., Fault detection and isolation using multiple Takagi-Sugeno fuzzy models, *Proc.of the 10th IEEE Int. Conf. on Fuzzy Systems*, Vol.3, 2001, pp.1498– 1502

[7] Ichtev, A., Hellendoom, J., Babuska, R.; Mollov, S., Fault-tolerant model-based predictive control using multiple TS fuzzy models, *Proc. of IEEE Int. Conf. on Fuzzy Systems*, Vol.1, 2002, pp. 346 – 351

[8] EN Skoundrianos, SG Tzafestas, Modeling of FDI of discrete time systems using a MLP with new sigmoidal activation function, *J. of Intelligent & Robotic Systems: Theory and applications*, Vol.41, No.1, 2004, pp.19-36

[9] H. A. Talebi, R.V. Patel, H. Asmer, NN based dynamic modeling of flexible link manipulators with application to the space station remote manipulator system, *J. of Robotic Systems*, Vol.17, No.7, 2000, pp.385-401

[10] Tinos R., Terra M., Bergerman M.,Fault tolerance in cooperative manipulators, *Proc. of IEEE Int. Conf. on Robotics & Automation*, 2002, Vol., pp.470-475

[11] K. Iagnemma and S. Dubowsky, Mobile robot rough-terrain control for planetary exploration, *ASME Biennial Mechanisms & Robotics Conf.*, MD, 2000.

[12] D.M. Helmick, Y. Chang, S.I. Roumeliotis, D. Clouse, and L. Matthies, Path following using visual odometry for a mars rover in high-slip environments, *Proc. of IEEE Aerospace Conf.*,Vol.2,2004,pp.772-789

[13] E. T. Baumgartner, H. Aghazarian, and A. Trebi-Ollennu, Rover localization results for the FIDO rover, *Proc. of Sensor Fusion Decentralized Control Auton. Robotic Syst Conf.*, 2001, pp. 34–44.

[14] C. Olson, L. Matthies, M. Schoppers, and M. Maimone, Robust stereoego-motion for long distance navigation, *Proc. IEEE Conf. on Comput.Vision Pattern Recognition*, Vol. 2, 2000, pp. 453–458.

[15] E. Tunstel, A. Howard, and H. Seraji, Fuzzy rule-based reasoning for rover safety an survivability, *Proc. IEEE Int. Conf. on Robotics and Automation*, Vol.2, 2001, pp.1413-1420

[16] G. Reina, L.Ojeda, A. Milella, and J Borenstein, Wheel slippage and sinkage detection for planetary rovers, *IEEE/ASME Trans. On Mechatronics*, Vol.11, No.2, 2006, pp.185-195

[17] E. Albas, T. Arikan, C. Kuzkaya, In-process motor testing results using model based fault detection approach, *Electrical Insulation Conf. and Electrical Manufacturing & Coil Winding Conf.*,2001,pp.643-647

[18] K. Iagnemma, C Brooks, and S. Dubowsky, Visual, tactile, and vibration-based terrain analysis for planetary rovers , *Proc. IEEE Conf on Aerospace*, 2004

[19] Terra M.H., Tinos R., FDI in robotic manipulators via NNs – a comparison among three architectures for residual analysis, *J. of Robotic Systems* , Vol.18, No.7, 2001, pp.357-374